applied sciences

MDPI

Article

Network Traffic Characteristics and Analysis in Recent Mobile Games

Daekyeong Moon

Computer Engineering Department, Myongji University, 116, Myongji-ro, Cheoin-gu, Yongin-si 17058, Gyeonggi-do, Republic of Korea; dkmoon@mju.ac.kr

Abstract: The landscape of mobile gaming has evolved significantly over the years, with profound changes in network reliability and traffic patterns. In the early 2010s, mobile games faced challenges due to unreliable networks and primarily featured asynchronous gameplay. However, in the current era, modern mobile games benefit from robust network connectivity, mirroring PC gaming experiences by relying on persistent connections to game servers. This shift prompted us to conduct an in-depth traffic analysis of two mobile games that represent opposite ends of the genre spectrum: a massively multiplayer game resembling PC MMORPGs with tightly synchronized gameplay, and a single-player puzzle game that incorporates asynchronous social interactions. Surprisingly, both games exhibited remarkably similar traffic footprints; small packets with short inter-packet arrival times, indicating their high expectations for network reliability. This suggests that game developers now prioritize network quality similarly to their PC gaming counterparts. Additionally, our analysis of packet lengths unveiled that recent mobile games predominantly employ short packets dominated by a few key packet types closely tied to player actions, which conforms to observations from PC online games. However, the self-similarity in traffic patterns, a notable feature in PC online games, only partially explains the traffic in mobile games, varying across genres. These findings shed light on the evolving traffic patterns in mobile games and emphasize the need for further research in this dynamic domain.

Keywords: mobile games; traffic analysis; Internet measurement

Citation: Moon, D. Network Traffic Characteristics and Analysis in Recent Mobile Games. *Appl. Sci.* **2024**, *14*, 1397. https://doi.org/10.3390/app14041397

Academic Editors: Chuan-Ming Liu and Wei-Shinn Ku

Received: 30 December 2023
Revised: 30 January 2024
Accepted: 6 February 2024
Published: 8 February 2024

1. Introduction

Network traffic analysis reports have illustrated that gaming is one of the most popular Internet applications, accounting for an estimated 8–10% of total Internet traffic and ranking as the third biggest traffic source after video streaming and web applications, including social media [1–3]. Another report highlights that 92.3% of Internet users access the Internet using mobile phones, with gaming being the most common use for these devices [4]. Given that game servers are typically hosted in cloud data centers and that game service providers incur significant costs for network traffic, understanding mobile game traffic patterns is crucial, not only from an engineering viewpoint but also from a business perspective.

As mobile devices, including smartphones and tablets, have become the prevailing service environment for the gaming industry, game genres that run on mobile devices have also become diverse and complicated. In the early era of networked mobile gaming in the 2010s, most mobile games were asynchronous due to the poor robustness and high cost of mobile networks. In these games, players could interact with other players, such as family members and friends, but they generally played independently rather than tightly together.

On the other hand, today's mobile network technology, especially in metropolitan areas, has significantly improved in connection stability. A recent survey showed that from October 2022 through March 2023, 5G and 4G networks in the UK exhibited 98.4% and 97.8% connection success rates on average, respectively, when a mobile device becomes active [5]. This suggests that mobile games enjoy much more robust network connectivity,

even in the mobile network, and thus such games now rely more on a persistent network connection to the game server. This phenomenal change suggests that the current underlying network traffic patterns might differ from those of the 2010s and from those in the PC gaming environment.

This paper illustrates traffic analysis from two mobile games with global service. The analyzed games represent opposite extremes in terms of their genre. One is strongly synchronized and a massive multiplayer game similar to that of a PC environment, but with an "auto-hunt" feature to allow gameplay without human engagement. The other is less synchronized and primarily played by a single player, although it also features social interactions among players.

The primary contribution of our work is demonstrating that recent mobile games exhibit traffic patterns akin to those in PC games, particularly regarding packet length and inter-packet arrival times, but have differences regarding traffic self-similarity. Our analysis of the games' traffic traces reveals that current mobile games typically feature notably short inter-packet arrival times, similar to traditional multiplayer games in the PC environment, accompanied by a long-tail distribution due to the intermittent sleep and resume nature of mobile applications. Concerning packet lengths, recent mobile games predominantly use shorter packets, favoring them over aggregated larger packets to conserve network bandwidth. These patterns, observed consistently across different game genres, suggest that modern mobile games are developed with expectations of reliable network connectivity, a notable shift from earlier in the 2010s. This consistency in traffic characteristics, irrespective of game genre, underscores a fundamental similarity in network usage between contemporary mobile and PC gaming platforms. Our analysis, however, discovered that the presence of self-similarity in traffic patterns, identified by previous research on PC online games' traffic patterns, varies across game genres. This finding encourages further research to model mobile game traffic.

This paper is organized as follows: Section 2 reviews research related to the mobile gaming industry. Section 3 briefly introduces the mobile games analyzed and explains the methods for collecting and anonymizing traffic data. In Section 4, we describe our analysis methodology and present key findings. Finally, Section 5 summarizes these findings and proposes directions for future research.

2. Related Work

Reports have shown that gaming is the most popular activity among mobile device users. Furthermore, gaming activities, which include those on PCs, mobile devices, and consoles, collectively constitute the third largest source of total Internet traffic [1–4]. Despite its significance in the context of today's Internet traffic, there is a notable dearth of comprehensive research on mobile game traffic. In contrast, considerable research efforts have been directed towards analyzing PC online game traces, modeling traffic patterns, and optimizing traffic [6–13]. Previous research agrees that PC online games generate highly periodic bursty short packets. In particular, Chen et al. discovered traffic exhibits pronounced periodicity and temporal locality in inter-packet arrival times, attributable to player action patterns by a comprehensive traffic analysis using substantial packet traces from a PC MMORPG game [11]. Feng et al. also observed that the distribution of game session time in PC games is not heavy-tailed, a characteristic stemming from the synchronized nature of gameplay [12]. Henderson et al. explored the network quality of service (QoS) tolerance of game players [14]. Apart from previous research mainly focusing on PC online games, our work fills the gap for mobile games. We confirmed that mobile games share similarities with PC online games regarding traffic patterns while having unique differences at the same time. Chen et al. conducted a preliminary traffic analysis using Pokémon Go, which is a mobile AR game [15], but their preliminary results did not uncover relationships with PC online games.

There is considerable research measuring the real-life performance of various mobile network technologies (i.e., the 3G, 4G, and 5G) [5,16–23]. Based on these research findings,

mobile application developers can reasonably assume that even 4G LTE, currently the most prevalent mobile network worldwide, can provide an average round-trip time (RTT) of about 50 ms between a carrier network and a player's device in metro areas. In particular, Ref. [22] reports that a study conducted in London showed a 5-millisecond latency could be achieved with 99.999% reliability over a 5G network. This research indicates that applications, including games, running on mobile networks can reliably expect lower latency as mobile infrastructure evolves. The finding in mobile network robustness is the main motivation for our research on mobile game traffic patterns.

Kämäräinen et al. investigated factors contributing to end-to-end latency in cloud gaming, where game scenes are rendered at cloud servers. Their research highlights emerging technologies that could enhance gaming experiences demanding high network bandwidth and strict latency constraints [24]. Similarly, Braud et al. conducted research on mobile AR applications, which require computational offloading to cloud servers and are, therefore, also subject to network bandwidth and latency limitations [25]. Despite the disruptive approach of cloud gaming, it was not widely deployed in the mobile gaming industry because of high costs and the limited robustness of mobile networks. This approach is, however, being revitalized by the rapid emergence of virtual reality technologies and technological improvements in mobile networks. Therefore, understanding mobile traffic characteristics through our research can also contribute to cloud gaming.

3. Materials and Methods

Due to the exclusivity or closed nature of the gaming industry, game traffic traces are not readily accessible, even in anonymized form. Game companies hesitate to release traffic traces not only because of concerns over players' privacy information but also due to fears that the data might be used to compromise the security of their services. This partially explains why there has been limited research conducted on the network traffic characteristics of online/mobile games, despite their significance in terms of traffic usage.

For this limitation, we selected two representative game genres at opposite ends of the spectrum and collected network traffic traces from a mobile game for each genre. The games are globally serviced and independently developed and operated by different game companies. Therefore, the games do not share any specific development methodology or assumptions about underlying network behavior.

3.1. Target Games

Table 1 summarizes the characteristics of the games used for analysis. We refer to each game as *A* and *B*, respectively, as the companies requested anonymity.

Table 1. Evaluated games and play characteristics.

Game	Genre	Play Pattern
A	MMORPG	Thousands of players share a single world. They cooperate and compete for rewards. Game offers "auto-hunt" feature to allow play without human engagement.
B	Puzzle	Individual players independently solve puzzles. Players can ask for help from others and visit their virtual private spaces. Players may receive hints for the puzzle, but are not allowed "auto" play.

Game *A* is of the massively multiplayer role-playing game (MMORPG) genre. It hosts a single medieval-style world shared by thousands of players. Players hunt monsters in dungeons in order to gain experience points and game items. This means that players sometimes cooperate and sometimes compete for game rewards. In this game style, even an idle game client receives network packets because of the activities of other players who are visible to the game client. Players have the flexibility to leave their characters in safe locations, like villages, or to actively participate in more challenging settings such

as monster dungeons. A distinctive feature of Game *A*, particularly setting it apart from traditional PC MMORPGs, is its "auto-hunt" function. This allows for automatic character engagement in dungeon hunting, enabling a more passive gameplay experience. Such a feature is increasingly common in modern mobile MMORPGs, reflecting the evolving nature of mobile gaming experiences.

Game *B* is a match-three puzzle game. Each player independently solves puzzles in various scenario settings by matching the same 3 tiles. The game is a single-player game by nature. It, however, added multiplayer features such as asking other players for help and decorating each player's own virtual space which is visible to other players. Players can interact in such ways, but not in the synchronized fashion. This genre has been one of the most popular mobile game genres since the first mobile game. In this genre, idle players — those connected to the game service but not actively engaging — do not generate network traffic. This is a significant divergence from MMORPGs like Game *A*, where even idle players continue to receive network packets.

3.2. Network Characteristics of the Evaluated Games

Table 2 describes the network protocols and application message formats utilized by the games.

Table 2. Network protocols and message formats.

Game	Protocol	Description
A	TCP	Application messages are in Protocol Buffers [26] on TCP.
B	HTTP/1.1 and /2	Designed to use RESTful API. Application messages are in JSON over HTTPS (using TCP).

3.3. Traffic Measurements

We conducted measurements from Game *A* and *B* servers, and Table 3 presents an overview of the measurements. We elaborate detailed information about the measurements in the following subsections.

Table 3. Key aspects about traffic measurements.

	Game *A*	Game *B*
Time span	2 h (10 July 2023 UTC)	2 days (5–6 August 2023 UTC)
Client-to-server messages	19,010,841	335,145,778
Server-to-client messages	16,917,007	335,145,778
Observed sessions	408	53,749,009
Session duration(s) (avg/med/stdev) [1]	1819.1/1305.2/2359.4	1563.7/629.4/2382.1
Msgs per session (avg/med/stdev)	49,540.8/15,429/86,622.8	448.2/210/478.9
Bit rates (KBps) (avg/med/stdev)	240.2/237.3/30.8	14,219.9/13,977.2/3648.0
Log size in total	5 GB	150 GB

[1] Due to the potential session terminations of a mobile application, we measured the time interval between the first and last message in a session instead.

3.3.1. Game *A* (MMORPG)

The game consists of tens of EC2 virtual machines (VMs). Each VM hosts a custom TCP server software (https://medium.com/@dpinoagustin/custom-tcp-application-with-golang-427998c8a75c (accessed on 23 January 2024)). Each server is assigned a particular space in the shared virtual space (e.g., one might correspond to a village, while another might host a monster dungeon and/or a battlefield). Therefore, the client directly communicates with a particular server depending on the player's location. This means that certain

measurements can be influenced by spatial aspects because players may show different behavioral patterns in different (virtual-world) locations.

In order to minimize such a spatial impact, we collected log messages from a server that is in charge of both a village and a monster dungeon. Each log entry records an application-level Protocol Buffers message as well as a timestamp, a message direction, a message type, a player session, and a message length. The measurement spans two hours during the high season, allowing us to capture traffic behavior during the period of peak activity. As previously mentioned, Game *A* offers an auto-hunting feature, which allows the player's character to engage in game activities even when the player is not actively controlling them. Consequently, this feature leads to longer session durations and an increased number of messages per session, as reflected in Table 3. Below is a part of the log message given:

3.3.2. Game *B* (Single-Player Puzzle)

The game is designed around the REST architecture. Thus, the game client invokes RESTful APIs over HTTP/1.1 and HTTP/2, provided by the game server. In fact, the server and the client communicate over a secured layer (i.e., HTTPS), but this detail is not directly relevant to our analysis. In such a design, the game server is stateless because each RESTful API invocation carries sufficient information for the game server to process the client's request. Therefore, it is common practice to place an HTTP load balancer between the game client and the game server for better server scaling and handling failures. Game *B* uses hundreds of EC2 VMs along with AWS Elastic Load Balancer (ELB) with the auto-scaling feature enabled. Because the game client can be directed to any of the game servers, depending on the requested REST API, by the ELB, we collected aggregated log messages at the ELB, rather than from a particular game server. Logs are in a format similar to standard HTTP access logs, and therefore, they include information such as request timestamp, request size, RESTful API method/URL, response status code, response size, response latency, etc. [27].

3.4. Trace Anonymization

Game *A* carries session information in its application messages. A new session identifier is assigned when a player authenticates themselves with the server. This includes cases where the mobile game application wakes up from an extended period of inactivity, as well as situations where a new player connects to the game servers. Although the session identifier used by Game *A* is in the UUID form, it is already an ephemeral, opaque handle that identifies a matching player and does not reveal any privacy information. Thus, the company did not perform any additional anonymization process, as illustrated in Listing 1.

Listing 1. Game *A*'s log example.

```
I0710 20:00:11.553869 12631 session_message_utils.cc:861] [C->S] TCP/Protobuf: type=_INT#20304, length=83, session=faa78195-da54-4f09-8472-cfb6b4754230
I0710 20:00:11.553946 12630 session_message_utils.cc:861] [S->C] TCP/Protobuf: type=_INT#20111, length=18, session=faa78195-da54-4f09-8472-cfb6b4754230
I0710 20:00:11.553989 12630 session_message_utils.cc:861] [S->C] TCP/Protobuf: type=_INT#20306, length=55, session=eb06d482-63b2-4a5e-86ec-d3ff6de165d7
I0710 20:00:11.554016 12630 session_message_utils.cc:861] [S->C] TCP/Protobuf: type=_INT#20318, length=23, session=faa78195-da54-4f09-8472-cfb6b4754230
I0710 20:00:11.554039 12630 session_message_utils.cc:861] [S->C] TCP/Protobuf: type=_INT#20305, length=20, session=faa78195-da54-4f09-8472-cfb6b4754230
I0710 20:00:11.554841 12629 session_message_utils.cc:861] [C->S] TCP/Protobuf: type=_INT#20110, length=71, session=ec1b9625-60cd-41ef-ba34-4513dc0fa3d0
```

On the other hand, the logs of Game *B* are in a format similar to standard HTTP access logs. Because these logs reveal source/destination IP addresses, they might disclose hints about the internal service architecture as well as the players' geographical locations. Therefore, the company hashed the IP addresses using SHA-1 before providing the log files. The game service runs with the HTTP persistent connection feature enabled. For the game, we used a pair of a hashed IP address and a port number to identify an active player. Below is a part of the log files demonstrating anonymized IP addresses (Listing 2):

Listing 2. Part of Game *B*'s log showing hashed IP addresses (lines are truncated for length).

```
h2 2023-08-05T23:50:00.824295Z - 31e5b1d68be657bbc1e5c90bb5c5db5f11229413:58070 4da5b696fbe39d9fea9d66950e98d7a4c0886027:8080 0.001 0.018 0.000 200 ...
https 2023-08-05T23:50:00.874779Z - ae9331a530340dc2e04b3e8cfd5127190f23747f:17204 5f4b3163bc19fb6338e25e2d4b0fa07a7b381cb4:8080 0.001 0.004 0.000  ...
h2 2023-08-05T23:50:00.899088Z - 31e5b1d68be657bbc1e5c90bb5c5db5f11229413:58070 aeae5ffbba3ed3b7e521fa3743f82a217e9c8d79:8080 0.001 0.002 0.000 200 ...
h2 2023-08-05T23:50:00.900988Z - 31e5b1d68be657bbc1e5c90bb5c5db5f11229413:58070 4da5b696fbe39d9fea9d66950e98d7a4c0886027:8080 0.001 0.002 0.000 200 ...
h2 2023-08-05T23:50:00.901045Z - 31e5b1d68be657bbc1e5c90bb5c5db5f11229413:58070 9c6332b9ad2e2e3fe2a754616694f5ee07edfa98:8080 0.001 0.004 0.000 200 ...
h2 2023-08-05T23:50:00.901939Z - 31e5b1d68be657bbc1e5c90bb5c5db5f11229413:58070 76f6ce7140f4bc04fa82598196d87879d6d38877:8080 0.001 0.006 0.000 200 ...
```

3.5. Log Processing

Due to the large size of the logs and the numerous player sessions, we first grouped the log files by the player's session identifier. For Game *A*, we used the session identifier information in the log file as it is, while for Game *B*, we employed a tuple of a hashed IP address and a port number. This approach not only helped in organizing the data more efficiently but also ensured the preservation of session continuity and player behavior patterns for accurate analysis.

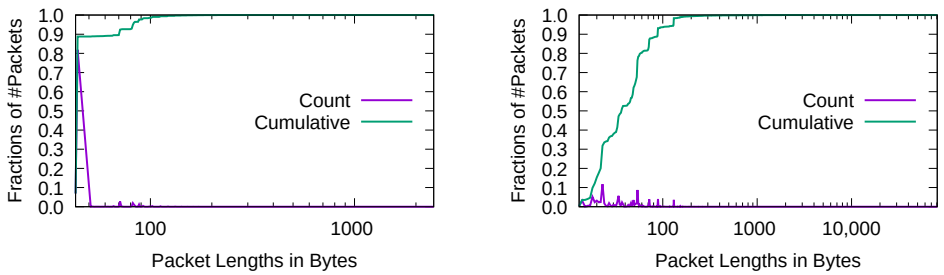
To analyze packet lengths, we classified them by traffic direction and accumulated the statistics over all player sessions. This classification allowed us to observe the differences in data flow between incoming and outgoing traffic, providing insights into the server–client interaction dynamics. For packet inter-arrival times, we measured the inter-arrival time for each direction per player session, and then computed the statistics per direction. Further, we segmented Game *B*'s data based on different time frames in 1 h granularity and game events to understand how player behavior and server responses varied under different conditions. This temporal analysis was pivotal in understanding the scalability of the game servers during peak and off-peak hours. Lastly, we organized packet arrivals into 100-millisecond time slots and used this information to create variance–time plots by calculating variances as the bucket aggregation level increased.

The analyzed results are presented and discussed in the following section. The source code used for log processing is available through the author's GitHub repository: https://github.com/dkmoon/2023trace_analysis (accessed on 7 Feburuary 2024). This repository includes all the scripts and methodologies used, ensuring transparency and reproducibility of our analysis.

4. Results and Discussion

4.1. Packet Lengths

Each of Game *A*'s log entry records a corresponding application message (i.e., Protocol Buffers message) length. Thus, it does not count the lengths of the IP header and the TCP header. On the other hand, Game *B*'s log is generated by the AWS ELB and it includes an HTTP header as well. Figures 1 and 2 illustrate the distributions of measured packet lengths. Because an HTTP header is an inevitable part of network traffic, for games like Game *B*, to carry application messages, we did not exclude the HTTP header length. Also, this approach helps us compare the packet lengths at the TCP-level regardless of the underlying networking topology.

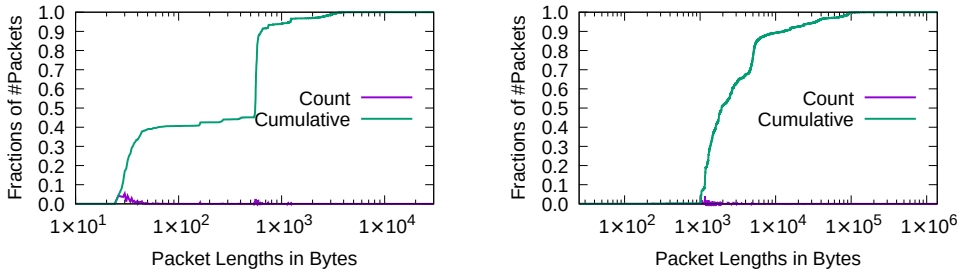


(**a**) Client-to-server traffic (**b**) Server-to-client traffic

Figure 1. Packet lengths distribution for Game *A* (mobile MMORPG). The graphs illustrate the lengths of data in TCP payload.

According to Game *A*'s measurement, mobile MMORPGs are observed to use very short packets (Figure 1); regardless of traffic direction, most application messages are less than or equal to 100 bytes long. Considering traffic direction, server-to-client traffic pattern reveals a much longer tail, while client-to-server traffic shows a handful of packets are dominant. In the measurement, 81.9% of client-to-server packets were 44 bytes long. It is because some of the server-to-client packets aggregate information such as a set of items or plenty of players, whereas client-to-server packets are dominated by the most frequent player activities. Indeed, the most outstanding 44 bytes long packet was about player move in our measurement. It is noteworthy that these findings are consistent with previous research conducted on PC MMORPG [11]. Despite the difference between wired connection and wireless connection, such a particular game genre that requires a strongly synchronized setting shows a similar traffic pattern. This may also indicate that the robustness of mobile networks has helped game developers safely assume robust connectivity between game client and game server.

Game *B* exhibits similar patterns in both client-to-server and server-to-client traffic: (1) client-to-server traffic is characterized by a small number of dominant packet types closely tied to player behavioral patterns, and (2) server-to-client traffic displays a longer-tail distribution. Figure 2 displays the packet length distributions for both types of traffic. It is important to note that these graphs include HTTP headers, resulting in overall packet lengths that are greater than those observed in games using direct TCP sockets (i.e., Game *A*). Considering the difference, Figure 2a demonstrates the prevalence of specific packet lengths, such as those around 500 bytes, similar to what was observed in Game *A*.



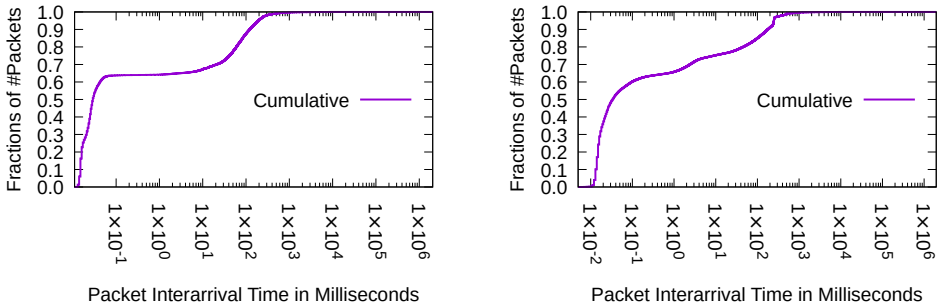
(**a**) Client-to-server traffic (**b**) Server-to-client traffic

Figure 2. Packet lengths distribution for Game *B* (single-player puzzle). Lengths also include HTTP headers. Thus, the graphs show the lengths of data in TCP payload.

4.2. Inter-Packet Arrival Times within a Session

We conducted an analysis of inter-packet arrival time as it serves as a valuable indicator of game developers' expectations regarding the robustness of the mobile network. Initially, we classified the collected network traffic by player session ID and subsequently calculated the inter-packet delay for each direction within each player session. For Game *B*, which uses request–response-style HTTP, we calculated a timestamp for each response according to a matching request timestamp, load-balancer delay, and server processing time. Figures 3 and 4 present the distributions of inter-arrivals computed through this process.

In the case of the MMORPG-style Game A, we observe remarkably short inter-packet arrival times (Figure 3). Approximately 99.6% and 99.5% of measured inter-arrival times fall within the 1000-millisecond threshold for both client-to-server and server-to-client traffic, respectively. This observation implies that the game places a high level of expectation on the quality of wireless network connectivity, irrespective of whether it is through mobile networks or Wi-Fi. This pattern is similar to those of PC games of the same genre, i.e., continuous interaction with players.

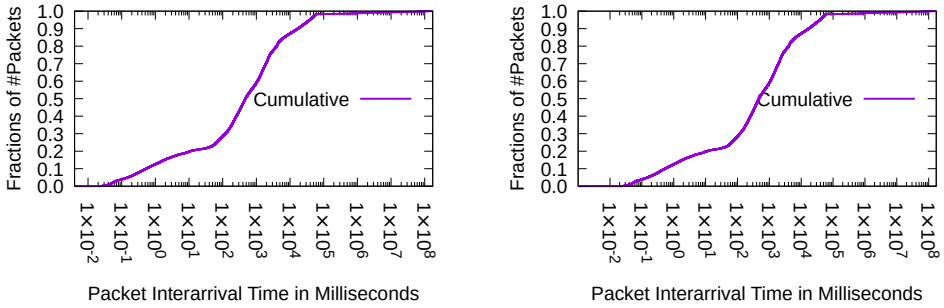


(**a**) Client-to-server traffic (**b**) Server-to-client traffic

Figure 3. Packet inter-arrivals distribution for Game *A*.

Meanwhile, in both directions, very long tails are revealed despite their short average inter-arrival times. Further analysis of packet types attributes this pattern to application wake-ups after extended periods of inactivity. This behavior is common among mobile applications, including games that frequently encounter interruptions and resume operations.

Game *B* also exhibits relatively short inter-arrival times in Figure 4. A significant 87.2% of the traffic falls within the 10,000-ms threshold. Notably, even when considering server processing time in the calculation of each response timestamp, the difference in distributions of requests and responses remain nominal. Given the gameplay patterns typical of puzzle games, the presence of 10-s inter-arrival times in each direction is notably short. Also, the distributions show a long tail similar to Game *A*. This implies that a mobile device's usage pattern has a significant impact on a mobile game's inter-packet arrival time, which was not observed from PC online games.



(**a**) Client-to-server traffic (**b**) Server-to-client traffic

Figure 4. Packet inter-arrivals distribution for Game *B*.

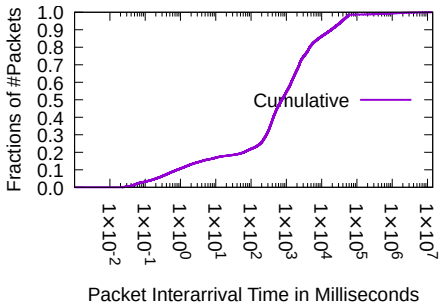
4.3. Temporal Difference

Interactions among players in a multiplayer game can result in increased network traffic, especially from the server to the client, making it essential to understand temporal differences in traffic patterns. In particular, we conjectured the inter-packet arrival time may be influenced by the concurrent number of connected players if the games are concerned about underlying network robustness and bandwidth. We split Game *B*'s log data into a 1-hour time frame and compared packet inter-arrival times for off-peak window and peak window. Table 4 summarizes the least busy time frame and the busiest time frame.

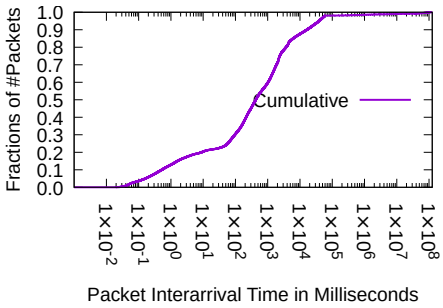
Figure 5 depicts the distribution of inter-arrival times for the compared time frames given in Table 4. Interestingly, the graphs exhibit similar patterns to those observed in Figure 4. Because the overall inter-packet latency is already noticeably short, as shown in Figure 4, this observation suggests that the mobile game does not employ any special traffic offloading techniques such as message aggregation. This may also indicate that recent mobile games have network performance expectations as high as those in PC MMORPGs.

Table 4. Statistics about the least busy time frame and the busiest time frame.

	Off-Peak	**Peak**
Time frame	20:00–21:00 UTC, 6 August	13:00–14:00 UTC, 5 August
Observed sessions	252,045	608,896
Server-to-client messages	3,507,062	8,638,416



(**a**) Off-peak traffic



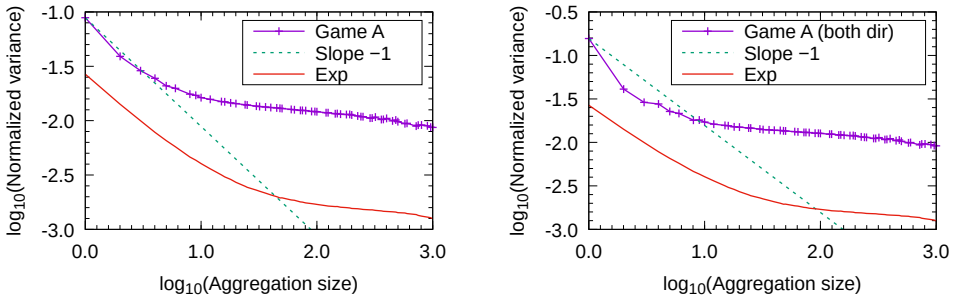
(**b**) Peak traffic

Figure 5. Distribution of packet inter-arrival times for Game *B* during off-peak and peak periods.

4.4. Self-Similarity

Previous research on PC MMORPG games has discovered that the distribution of packet arrival times in such game genres exhibits self-similarity; that is, it displays similar burstiness characteristics regardless of the time scale [11]. Therefore, we initially conjectured that mobile games would exhibit the same characteristics because they seem to have similar expectations of network robustness. Burstiness over different time scales can be assessed via a "variance–time plot" [28]. We first count the packets observed every 100 milliseconds. Then, we aggregate these 100-millisecond buckets and calculate the variance across the aggregated buckets by increasing the aggregation size. By plotting the aggregation sizes and corresponding variances on a log–log graph, we can check if the traffic exhibits self-similarity. If the graph exhibits a linear line with a slope between -1 and 0, it indicates the traffic pattern has a similar burstiness characteristic regardless of the time scale. For comparison between traffic with different numbers of arrivals, we normalized variances by dividing by the square of the mean of 100-millisecond buckets.

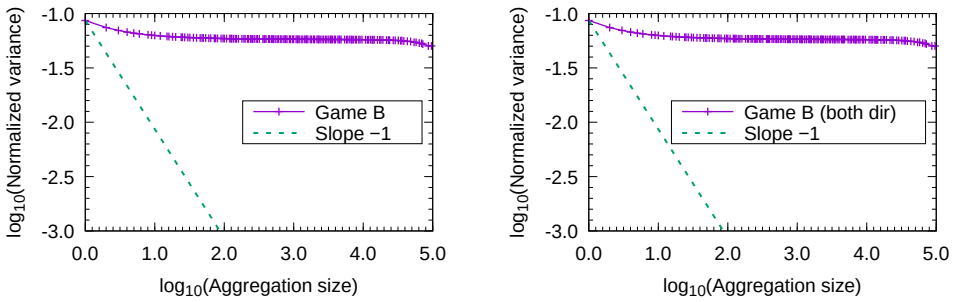
Figure 6 presents the variance–time plot for Game *A*. The dashed green line represents a reference line with a slope of -1 on the log–log scale, serving as a benchmark for comparing the self-similarity characteristics of the traffic pattern. The client-to-server traffic does not exhibit strict self-similarity; its variance decreases more steeply at lower time scales than at larger time scales, as shown in Figure 6a. This indicates that the traffic has self-similarity at a larger time scale (i.e., time scale more than 1 s), but its burstiness characteristic is different from that at a lower scale. To understand the traffic pattern, we compared the result to synthesized traffic (i.e., the solid yellow line labeled "Exp"). We generated traffic such that the arrival times for each session follow an exponential distribution with a mean rate of 1 packet per second. Then, we multiplexed 400 independent identically distributed (i.i.d.) sessions. The comparison suggests that the trace resembles an aggregated exponential process rather than a self-similar process. Including server-to-client traffic exaggerates the pattern and now clearly shows that the traffic does not have self-similarity because it shows a non-linear curve below the dashed line with a slope of -1 (Figure 6b). The results may be explained by traffic that is not directly initiated by the player (e.g., keep-alive traffic to detect mobile network hiccups or "auto-hunt" mode, which is typically adopted by a mobile MMORPG that plays a game on human's behalf), but further research is required to understand this correctly.



(**a**) Client-to-server traffic (**b**) Whole traffic

Figure 6. Variance–time plot for Game *A*. Both axes are on a logarithmic scale (base 10). The x-axis indicates how many 100-millisecond buckets are aggregated. The y-axis represents the variance of aggregated packet counts normalized by the square of packet counts per 100 milliseconds.

We conducted a similar analysis on the traffic trace of Game *B*. Figure 7a,b present the variance–time plots for the client-to-server traffic and the overall traffic, respectively. Both graphs exhibit similar patterns, which can be attributed to Game *B*'s use of HTTP protocol that operates on a request–response basis. In this setup, the server-to-client responses closely follow the corresponding client-to-server requests, typically after a short processing delay of less than 100 milliseconds. Consequently, including server-to-client traffic essentially doubles the packet arrivals in 100-millisecond buckets, but does not significantly alter the overall pattern of the graph. The linear and slow decay nature of both graphs suggests strong self-similarity in the traffic pattern. Contrary to Game *A*, the traffic in this type of game is strongly linked to player engagement, which partially explains the burstiness pattern observed.



(**a**) Client-to-server traffic (zoomed in) (**b**) Client-to-server traffic (zoomed out)

Figure 7. Variance–time plot for Game *B*. Both axes are on a logarithmic scale (base 10). The y-axis represents the variance of aggregated packet counts normalized by the square of packet counts per 100 milliseconds.

5. Conclusions

Initially, we hypothesized that different game genres would exhibit unique network traffic patterns and that mobile games would significantly differ from PC online games due to traditionally less reliable mobile networking environments. However, our analysis of two mobile games representing opposite genre spectrums suggests that modern mobile games maintain high expectations for network reliability, similar to PC online games, regardless of genre.

Contrary to our assumption that the wireless environment would influence packet lengths, our findings indicate that there are no substantial differences in packet lengths between recent mobile and PC online games. Both environments predominantly use short packets, closely linked to player action patterns, which aligns with findings from previous PC game research. Additionally, our analysis revealed that inter-packet arrivals within a

mobile game session are very short, resembling the patterns observed in PC online games. This suggests that modern mobile games do not implement packet aggregation, as they have high expectations for network robustness.

On the other hand, the presence of traffic self-similarity, commonly observed in PC online games, varies across mobile game genres. While the HTTP-based, less-synchronized game exhibits self-similarity in its traffic, the TCP-based, tightly synchronized game does not appear to have strict self-similarity. This observation contradicts our assumption that similar game genres (e.g., MMORPG) might share similar traffic characteristics. The difference may arise from variances in genre and/or player behavioral patterns between wired and wireless networks. While further research is needed for a more comprehensive understanding, there is a potential opportunity to devise a traffic model for mobile games that better suits their unique characteristics compared to the self-similar process observed in PC online games.

Despite the critical role of network traffic in the current Internet landscape, research on mobile game traffic patterns is surprisingly sparse. As mobile devices increasingly dominate the gaming sector, we advocate for more research in this area, emphasizing the importance of traffic analysis and the potential for developing predictive models for traffic patterns in mobile games.

Author Contributions: Conceptualization, D.M.; methodology, D.M.; software, D.M.; formal analysis, D.M.; data curation, D.M.; writing, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the 2022 Research Fund of Myongji University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from game companies that requested anonymity. The data are available from the author, subject to the companies' written permission. Source code used for analysis is available at https://github.com/dkmoon/2023trace_analysis (accessed on 23 January 2024).

Acknowledgments: Author acknowledges Minkyong Cho of Myongji University for his technical support in methodology development.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MMORPG	massively multiplayer role-playing game
AR	augmented reality
RTT	round-trip time
Game *A*	evaluated mobile game of the MMORPG style
Game *B*	evaluated mobile game of match-three style

References

1. Share of Internet Data Traffic Worldwide in the First Half of 2021, by Category. Available online: https://www.statista.com/statistics/1312357/global-data-traffic-by-content-type/ (accessed on 23 January 2024).
2. Distribution of Global Downstream Internet Traffic as of October 2018, by Category. Available online: https://www.statista.com/statistics/271735/internet-traffic-share-by-category-worldwide/ (accessed on 23 January 2024).
3. Amount of Data Created Daily (2024). Available online: https://explodingtopics.com/blog/data-generated-per-day (accessed on 23 January 2024).

4. Internet Traffic from Mobile Devices (Jan 2024). Available online: https://explodingtopics.com/blog/mobile-internet-traffic (accessed on 23 January 2024).
5. Mobile Matters (Published 20 July 2023). Available online: https://www.ofcom.org.uk/__data/assets/pdf_file/0018/264600/mobile-matters-2023-report.pdf (accessed on 23 January 2024).
6. Henderson, T.; Bhatti, S. Modelling User Behaviour in Networked Games. In Proceedings of the Ninth ACM International Conference on Multimedia, Ottawa, Canada, 30 September–5 October 2001; MULTIMEDIA '01, pp. 212–220. [CrossRef]
7. Färber, J. Network Game Traffic Modelling. In Proceedings of the 1st Workshop on Network and System Support for Games, Braunschweig, Germany, 16–17 April 2002; NetGames '02, pp. 53–57. [CrossRef]
8. Lang, T.; Branch, P.; Armitage, G. A Synthetic Traffic Model for Quake3. In Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Singapore, 3–5 June 2004; ACE '04, pp. 233–238. [CrossRef]
9. Dainotti, A.; Pescapè, A.; Ventre, G. A packet-level traffic model of Starcraft. In Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems, San Diego, CA, USA, 21 July 2005; pp. 33–42. [CrossRef]
10. Kim, J.; Choi, J.; Chang, D.; Kwon, T.; Choi, Y.; Yuk, E. Traffic Characteristics of a Massively Multi-Player Online Role Playing Game. In Proceedings of the 4th ACM SIGCOMM Workshop on Network and System Support for Games, Hawthorne, NY, USA, 10–11 October 2005; NetGames '05, pp. 1–8. [CrossRef]
11. Chen, K.T.; Huang, P.; Huang, C.Y.; Lei, C.L. Game Traffic Analysis: An MMORPG Perspective. In Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video, Stevenson, WA, USA, 13–14 June 2005; NOSSDAV '05, pp. 19–24. [CrossRef]
12. Feng, W.C.; Chang, F.; Feng, W.C.; Walpole, J. A Traffic Characterization of Popular On-Line Games. *IEEE/ACM Trans. Netw.* **2005**, *13*, 488–500. [CrossRef]
13. Saldana, J.; Navajas, J.; Ruíz-Mas, J.; Viruete-Navarro, E.; Casadesus, L. Online FPS games: Effect of router buffer and multiplexing techniques on subjective quality estimators. *Multimed. Tools Appl.* **2012**, *71*, 1823–1856. [CrossRef]
14. Henderson, T.; Bhatti, S. Networked Games: A QoS-Sensitive Application for QoS-Insensitive Users? In Proceedings of the ACM SIGCOMM Workshop on Revisiting IP QoS: What Have We Learned, Why Do We Care? Karlsruhe, Germany, 25–27 August 2003; RIPQoS '03, pp. 141–147. [CrossRef]
15. Chen, H.Y.; Hsu, R.T.; Chen, Y.C.; Hsu, W.C.; Huang, P. AR Game Traffic Characterization: A Case of Pokémon Go in a Flash Crowd Event. In Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual Event, 24 June 2021–2 July 2021; MobiSys '21, pp. 493–494. [CrossRef]
16. Serrano, C.; Garriga, B.; Velasco, J.; Urbano, J.; Tenorio, S.; Sierra, M. Latency in Broad-Band Mobile Networks. In Proceedings of the VTC Spring 2009—IEEE 69th Vehicular Technology Conference, Barcelona, Spain, 26–29 April 2009; pp. 1–7. [CrossRef]
17. Chetty, M.; Sundaresan, S.; Muckaden, S.; Feamster, N.; Calandro, E. Measuring Broadband Performance in South Africa. In Proceedings of the 4th Annual Symposium on Computing for Development, Cape Town, South Africa, 6–7 December 2013; ACM DEV-4 '13. [CrossRef]
18. Sommers, J.; Barford, P. Cell vs. WiFi: On the Performance of Metro Area Mobile Connections. In Proceedings of the 2012 Internet Measurement Conference, Boston, MA, USA, 14–16 November 2012; IMC '12, pp. 301–314. [CrossRef]
19. Nikravesh, A.; Choffnes, D.R.; Katz-Bassett, E.; Mao, Z.M.; Welsh, M. Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis. In Proceedings of the 15th International Conference on Passive and Active Measurement—Volume 8362, Los Angeles, CA, USA, 10–11 March 2014; PAM 2014, pp. 12–22. [CrossRef]
20. Becker, N.; Rizk, A.; Fidler, M. A measurement study on the application-level performance of LTE. In Proceedings of the 2014 IFIP Networking Conference, Trondheim, Norway, 2–4 June 2014; pp. 1–9. [CrossRef]
21. Albaladejo, M.B.; Leith, D.J.; Manzoni, P. Measurement-based modelling of LTE performance in Dublin city. In Proceedings of the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 4–8 September 2016; pp. 1–6. [CrossRef]
22. Al-Saadeh, O.; Wikstrom, G.; Sachs, J.; Thibault, I.; Lister, D. End-to-End Latency and Reliability Performance of 5G in London. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]
23. Braud, T.; Kämäräinen, T.; Siekkinen, M.; Hui, P. Multi-carrier Measurement Study of Mobile Network Latency: The Tale of Hong Kong and Helsinki. In Proceedings of the 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Shenzhen, China, 11–13 December 2019; pp. 1–6. [CrossRef]
24. Kämäräinen, T.; Siekkinen, M.; Ylä-Jääski, A.; Zhang, W.; Hui, P. A Measurement Study on Achieving Imperceptible Latency in Mobile Cloud Gaming. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei, Taiwan, 20–23 June 2017; MMSys'17, pp. 88–99. [CrossRef]
25. Braud, T.; Bijarbooneh, F.H.; Chatzopoulos, D.; Hui, P. Future Networking Challenges: The Case of Mobile Augmented Reality. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 1796–1807. [CrossRef]
26. Protocol Buffers Documentation. Available online: https://protobuf.dev/ (accessed on 28 December 2023).

27. Access Logs for Your Application Load Balancer. Available online: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html (accessed on 28 December 2023).
28. Leland, W.E.; Taqqu, M.S.; Willinger, W.; Wilson, D.V. On the self-similar nature of Ethernet traffic (extended version) *IEEE/ACM Trans. Netw.* **1994**, *2*, 1–15. [CrossRef]